

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Boyd H. TIMOTHY, et al.

Serial No.: 10/066,465 Examiner: Manglesh M. PATEL

Filed: January 30, 2002 Art Unit: 2178

Confirmation No.: 3498

For: METHOD AND APPARATUS TO DYNAMICALLY PROVIDE WEB
CONTENT RESOURCES IN A PORTAL

Date: February 28, 2007

Mail Stop Amendment
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

DECLARATION OF ARIEL S. ROGSON UNDER 37 C.F.R. § 1.131

I, ARIEL S. ROGSON, hereby declare:

1. I am the attorney of record in the present patent application. I am the attorney who drafted the patent application assigned to U.S. Patent Application Serial No. 10/066,465.

2. On October 4, 2001, our firm received invention disclosures IDR-532 and IDR-533 from Novell, Inc., assignee of this patent application by assignment recorded at Reel/Frame 012560/0451 (7 pages). Invention disclosure IDR-532 was titled "METHOD TO DYNAMICALLY DETERMINE A USERS LANGUAGE FOR THE INTERNET"; invention disclosure IDR-533 was titled "METHOD AND APPARATUS TO DYNAMICALLY PROVIDE WEB CONTENT RESOURCES IN AN INTERNET PORTAL". Invention disclosure IDR-532 was assigned to our docket number 6647-29; invention disclosure IDR-533

was assigned to our docket number 6647-30. Invention disclosures IDR-532 and IDR-533 were accompanied with a request to prepare and file patent applications with the United States Patent and Trademark Office for the invention disclosure by January 18, 2002. The interval between our receipt of the invention disclosures and the target filing date is a typical interval for Novell, Inc. A true copy of the pertinent portion of invention disclosure IDR-533 is attached hereto as Exhibit A.

3. When I received invention disclosures IDR-532 and IDR-533, I was in the process of completing other client work, and the preparation of the patent applications for IDR-532 and IDR-533 was put into my normal work queue based on the priority of the respective work.

4. I spoke with some of the inventors about these invention disclosures in early November 2001. Prior to the interview of the inventors, I reviewed the invention disclosures. During the interview, the inventors clarified my understanding about the invention disclosures.

5. During November 2001, I prepared patent applications for invention disclosures IDR-532 and IDR-533 based on the invention disclosures IDR-532 and IDR-533, as clarified by the interview with the inventors. Because these applications are related, each application references the other. On November 29, 2001, I sent draft patent applications for invention disclosures IDR-532 and IDR-533 to the client for review. I received feedback from the client between December 7, 2001 and December 21, 2001. On January 9, 2002, I sent a final draft to the inventors for their review before signing the declaration. On January 25, 2002, our firm received the executed declaration from the inventors and the executed Power of Attorney from the client.

6. On January 30, 2002, I filed the patent applications with the U.S. Patent & Trademark Office. Invention disclosure IDR-532, titled "METHOD TO DYNAMICALLY DETERMINE A USERS LANGUAGE FOR THE INTERNET" was assigned U.S. Application Serial No. 10/066,368; invention disclosure IDR-533, titled "METHOD AND APPARATUS TO

DYNAMICALLY PROVIDE WEB CONTENT RESOURCES IN AN INTERNET PORTAL”, was assigned U.S. Application Serial No. 10/066,465, which is the present patent application.

7. The claims of U.S. Patent Application Serial No. 10/066,465 are rejected under 35 U.S.C. § 103(a) over U.S. Patent Application Publication No. 2002/0174150 to Dang et al. (“Dang”) in view of non-patent publication “CSS Mobile Profile 1.0” to Wugofski et al. (“Wugofski”). Dang has an effective filing date in the United States of May 18, 2002. Wugofski has a publication date of October 24, 2001.

8. The claims of U.S. Patent Application Serial No. 10/066,465 are supported by the invention disclosure IDR-533. For example, the support for claims 1, 16, 31, 39, and 47 are described below (the remaining claims are similarly supported):

Claim 1. An apparatus for presenting content to a user, comprising:
a plurality of layout strings files;
a plurality of layout information files to describe how a layout string is displayed for a unique combination of a language and a device; and
a computer to store the layout strings files and the layout information files.

Claim 16. A computer-implemented method for displaying content to a user, comprising:
locating a layout information file from a plurality of layout information files specifying how a layout string is to be presented to the user for a unique combination of a language and a device;
locating one of a plurality of layout strings files storing the layout string; and
presenting the layout string to the user according to the located layout information file.

Claim 31. One or more computer-readable media containing a program to display content to a user, comprising:
location software to locate a layout information file from a plurality of layout information files specifying how a layout string is to be presented to the user for a unique combination of a language and a device;

location software to locate one of a plurality of layout strings files storing the layout string; and

presentation software to present the layout string to the user according to the located layout information file.

Claim 39. An article comprising:

a computer-readable modulated carrier signal;

means embedded in the signal for locating a layout information file from a plurality of layout information files specifying how a layout string is to be presented to a user for a unique combination of a language and a device;

means embedded in the signal for locating one of a plurality of layout strings files storing the layout string; and

means embedded in the signal for presenting the layout string to the user according to the located layout information file.

Claim 47. A computer-implemented method for using a selected context to display content to a user, comprising:

locating a layout information file from a plurality of layout information files specifying how the content is to be presented to the user for a unique combination of a language and a device;

locating a layout strings file storing a layout string in the selected context; and

presenting the content and the layout sting in the selected context to the user according to the located layout information file.

Each of the elements of the foregoing claims is supported in the specification of the present patent application as follows:

Referring to FIG. 8 of the original Patent Application drawings, a file hierarchy structure is shown including a plurality of layout strings files and a plurality of layout information files.

Page 12, lines 26-30 of the Patent Application describes the layout strings files of FIG. 8 as follows:

Main_lang.xml 855 and its siblings store the layout strings (described above with reference to FIGs. 4-5). That is, main_lang.xml 855 stores the default layout strings, whereas main_lang_en_US.xml (860), main_lang_en_UK.xml (865), and main_lang_es_ES.xml (870) store the American English, British English, and Spanish layout strings, respectively. A person skilled in the art will recognize there can be other layout string files as well.

Page 11, lines 25-28 of the patent application describes the layout information files (LIFs) of FIG. 8 as follows:

Skins 810 is the subdirectory storing information about each skin (i.e., LIF) available for the device. Each skin is assigned a name, one of which is Default 815. For example, in FIG. 8, Alternative Skin 1 875 is one alternative skin. Other alternative skins can also be present.

Page 6, lines 20-24 of the Patent Application describes how a layout information files can describe how a layout string is displayed for a particular language.

A person skilled in the art will recognize that there are times when LIFs can be language dependent. For example, some languages are displayed right-to-left (such as Japanese, Hebrew, or Arabic). For these languages, a LIF that has the information displayed in a different presentation can be preferable. How LIFs are associated with particular languages is discussed further with reference to FIG. 7A below.

Page 6, lines 5-10 of the Patent Application describes how a layout information file can describe how a layout string is displayed for a particular device.

FIG. 4 shows one of the gadgets of FIG. 2 with two different layouts, according to an embodiment of the invention. In FIG. 4, layout information file (LIF) 405 specifies one layout for gadget 210, and LIF 455 specifies an alternative layout for gadget 210. (A layout information file is sometimes called a *skin*.) For example, LIF 405 might represent the default layout for gadget 210, whereas LIF 455 might represent a layout of gadget 210 for portable devices (where space is at a premium).

Each of the elements of claims 1, 16, 31, 39, and 47 is also supported in Exhibit A as follows:

Figure 1, on page 2 of Exhibit A, shows a file hierarchy similar to the file hierarchy shown in FIG. 8 of the Patent Application, with a plurality of layout strings files and layout information files.

Pages 2-3 of Exhibit A describes Gadget Stylesheets and corresponds to the above description of layout information files from the patent application:

In the general case, a gadget will write one layout stylesheet for each skin (look) that it provides in the following directory:

<gadget>/skins/<skin name>/devices/<device>/

In our example (**Fig. 1**), we have created the stylesheet:

com.novell.nps.gadgets.GadgetX/skins/default/devices/default/main.xsl

Additional localized layout stylesheets may be defined using the localized file naming described above. This is useful in situations in which the portal must support multiple languages where the actual layout of a page must be different for two or more languages or locales. For example, if a portal needed to support both Japanese and English, the designer may wish to create a different layout than for Japanese users because Japanese is read left to right. If the original layout stylesheet is named *main.xsl*, this could be accomplished by defining the following layout stylesheet for Japanese:

<gadget>/skins/<skin name>/devices/<device>/main_jp_JP.xsl

To support the English portion that this portal would need to provide, the designer could either define an additional stylesheet for English in the same directory or allow the portal to default to the original *main.xsl* stylesheet.

(emphasis in original).

Page 3 of Exhibit A describes Gadget String Files and corresponds to the above description of the layout strings file of the patent application.

Novell Portal Services uses XSL Stylesheets to store the localized strings needed by gadget layout stylesheets. Inside each language stylesheet, a gadget must define globally unique XSL variables that will be referenced in the layout stylesheets.

Each gadget requires an XSL/Language XSL file pair to provide the correct language and locale information for each user who authenticates to the portal. A gadget should define a group of Language XSL files for each language it plans to support. These files should follow the same naming pattern described earlier in this document.

For a basic and simple implementation, these files should be created and stored in the gadget's *skins/default/devices/default/* directory. In our example gadget (**Fig. 1**), GadgetX has defined *main_lang_[language code]_[country code].xsl* in the directory:

com.novell.nps.gadgets.GadgetX/skins/default/devices/default/

As previously described for gadget stylesheets, the portal includes a mechanism of providing additional levels of granularity as needed.

One example of when this is needed is when the portal must support both large and small display devices. It may be desired to provide detailed descriptions in Spanish when a user is using Internet Explorer 5 on their desktop computer, but short explanations when they login using a PocketPC device. In this scenario, two files with the same name are required in the following directories:

<gadget>/skins/<skin name>/devices/ie5/main_lang_es_ES.xml

<gadget>/skins/<skin name>/devices/pocketpc/main_lang_es_ES.xml

(emphasis in original).

Thus, invention disclosure IDR-533 discloses a plurality of layout information files, a plurality of layout string files, and presenting a layout string according to a layout information file, and therefore supports claims 1, 16, 31, 39, and 47. The remaining claims are similarly supported by invention disclosure IDR-533.

9. Because invention disclosure IDR-533 supports the claims in U.S. Patent Application Serial No. 10/066,465, the invention was conceived before October 24, 2001, the publication date of Wugofski, and diligently embodied in the present patent application filed January 30, 2002.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.



Ariel S. Rogson

Dated: 28 Feb 2007

Novell Portal Services Dynamic Look and Feel Architecture

Abstract

The portal provides the ability to obtain the user's desired language and locale information from the directory. This allows dynamic support of users' language and locale in portal web pages. Using the language and locale information found in the directory, the portal's main stylesheets and each gadget's stylesheet will be dynamically selected. Using this same mechanism, the correct layout (skin) and device resource will be used in building the portal web page for any given user.

Gadget Directory Overview

Fig. 1 is a typical gadget's directory structure in Novell Portal Services. We recommend this same structure for every gadget that will be built for use in the portal. This structure enables the following:

1. Localization of the strings used in the gadget's layout stylesheet
2. Localization of the layout stylesheets themselves
3. Ability for a gadget to support different skins

One of the goals in the localization of each gadget's strings is to eliminate file duplication. That is to say that many different layout stylesheets should be able to utilize the same string file.

It is possible, however, for each gadget to define multiple string files if necessary. This will be useful, for example, when a gadget desires to provide a lengthy description for large screen devices and brief descriptions for small screen devices like a PDA.

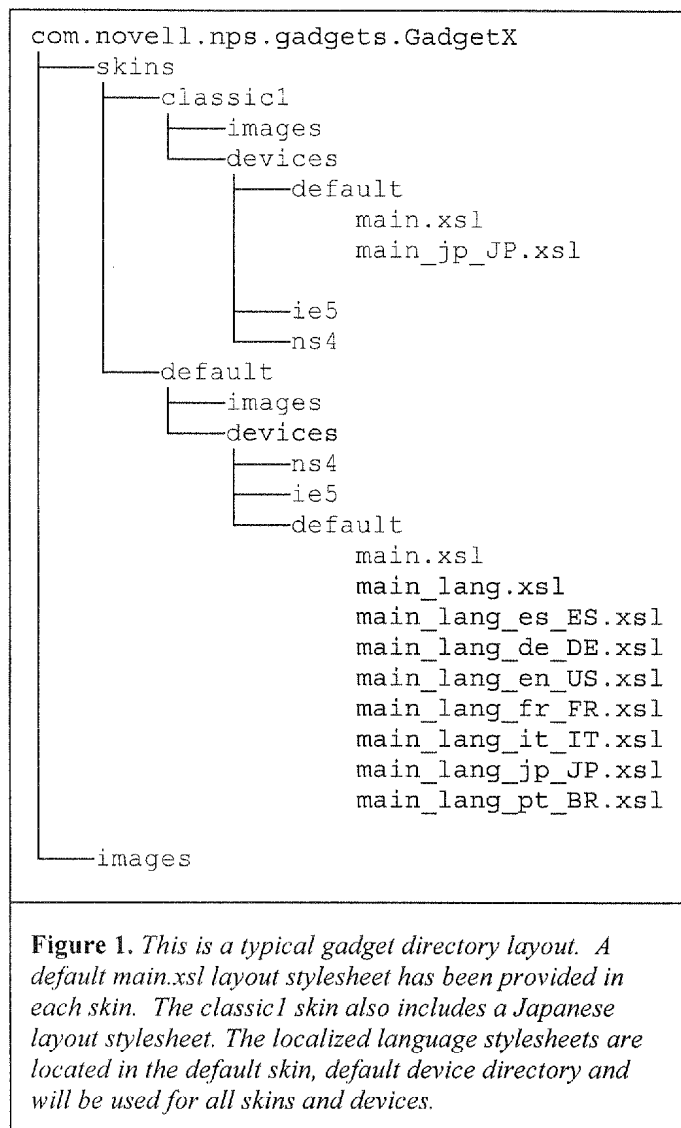
Note: The portal's system stylesheets directory will follow the same directory structure outlined in this document for managing the different skins and localized files.

Localized File Naming

Throughout this document, we will refer to the localized naming of files. Novell Portal Services adheres to the ISO 639 Standard for language codes and ISO 3166 Standard for country codes for naming localized files. The portal will automatically use localized layout and language stylesheet files when available. These files will be named using the following pattern:

[non-localized filename]_[language code]_[country code].[extension]

For example, the following filename would be created when localizing *main.xsl* for the English language in the United States:



main_en_US.xml

The same pattern is used for all localized files for different languages and countries.

Gadget Stylesheets

In the general case, a gadget will write one layout stylesheet for each skin (look) that it provides in the following directory:

`<gadget>/skins/<skin name>/devices/<device>/`

In our example (**Fig. 1**), we have created the stylesheet:

com.novell.nps.gadgets.GadgetX/skins/default/devices/default/main.xml

Additional localized layout stylesheets may be defined using the localized file naming described above. This is useful in situations in which the portal must support multiple languages where the actual layout of a page must be different for two or more languages or locales. For example, if a portal needed to support both Japanese and English, the designer may wish to create a different layout than for Japanese users because Japanese is read left to right. If the original layout stylesheet is named *main.xml*, this could be accomplished by defining the following layout stylesheet for Japanese:

```
<gadget>/skins/<skin name>/devices/<device>/main_jp_JP.xml
```

To support the English portion that this portal would need to provide, the designer could either define an additional stylesheet for English in the same directory or allow the portal to default to the original *main.xml* stylesheet.

Gadget String Files

Novell Portal Services uses XSL Stylesheets to store the localized strings needed by gadget layout stylesheets. Inside each language stylesheet, a gadget must define globally unique XSL variables that will be referenced in the layout stylesheets.

Each gadget requires an XSL/Language XSL file pair to provide the correct language and locale information for each user who authenticates to the portal. A gadget should define a group of Language XSL files for each language it plans to support. These files should follow the same naming pattern described earlier in this document.

For a basic and simple implementation, these files should be created and stored in the gadget's *skins/default/devices/default/* directory. In our example gadget (**Fig. 1**), GadgetX has defined *main_lang_[language code]_[country code].xml* in the directory:

```
com.novell.nps.gadgets.GadgetX/skins/default/devices/default/
```

As previously described for gadget stylesheets, the portal includes a mechanism of providing additional levels of granularity as needed.

One example of when this is needed is when the portal must support both large and small display devices. It may be desired to provide detailed descriptions in Spanish when a user is using Internet Explorer 5 on their desktop computer, but short explanations when they login using a PocketPC device. In this scenario, two files with the same name are required in the following directories:

```
<gadget>/skins/<skin name>/devices/ie5/main_lang_es_ES.xml  
<gadget>/skins/<skin name>/devices/pocketpc/main_lang_es_ES.xml
```

The default device and default skin

The *devices/default* and *skins/default* directories are default implementations. When the portal fails to find localized stylesheets or string files in a given device/skin directory, it will default to use the ones found in the *devices/default* or *skins/default* directories. If the portal cannot find the needed file in these default directories, the process will fail and needs to be fixed by the administrator. This process will be described in more depth later.

Managing Image Files

Each gadget contains an *images* directory that can be used to store images specific to the gadget. In addition, each skin within a gadget also has an *images* directory. This should be used for graphics specific to a particular skin. It is the responsibility of the person writing the gadget layout stylesheet to reference the images in their proper location.

Gadgets may use localized image files. By this, we mean that an image can contain a graphic which is either specific to a specific language, locale or both. In this case, the gadget writer is responsible for creating and referencing the different graphic files for the different languages and locales. To maintain organization, a gadget writer can adopt the same ISO standards for languages and country codes to name graphics.

To avoid creating multiple layout stylesheets, a gadget writer should create variables in the language stylesheet files that reference the correct localized images. As an example of this, a variable to a “stop” icon could be created:

```
<xsl:variable name="com.novell.nps.GadgetX.images.StopIcon">
  <path to gadget's images>/stop_icon_fr_FR.jpg
</xsl:variable>
```

In the gadget stylesheet, this can be referenced using an XSL Attribute Value Template:

```

```

Locating Localized Files

When a user first accesses Novell Portal Services, the portal will determine what language to use based on the browser's language. When the user authenticates to the Portal, the language and locale information will come through a prioritized list that is stored in the user's object in the directory.

Using the user's language information and a routine in the portal, gadget writers will be able to ask for a localized version of a resource file. In addition, the portal will provide the necessary mechanisms to dynamically build the correct set of layout and language stylesheet files to provide the user with the correct language on their device.

The search routine is built upon an interface that allows different search strategies to be selected by the administrator. By default, the portal will search the directory structure for localized files in the following order (default directories have been underlined):

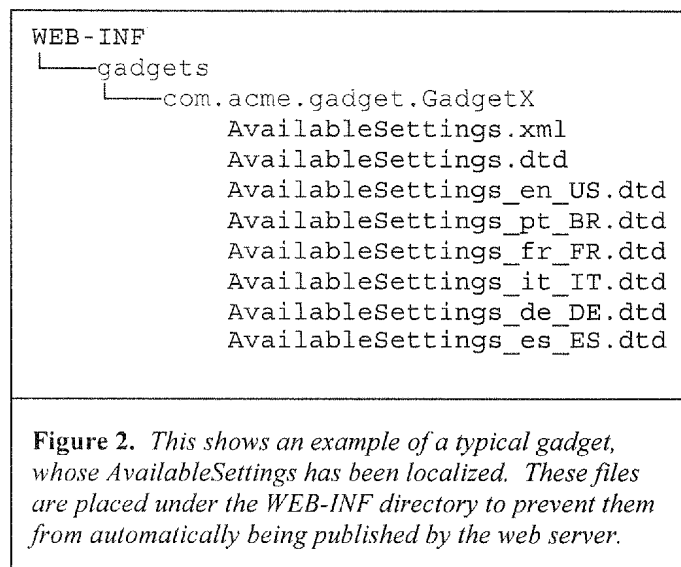
1. `<gadget>/skins/<skin name>/devices/<device>/`
2. `<gadget>/skins/<skin name>/devices/default/`
3. `<gadget>/skins/default/devices/<device>/`
4. `<gadget>/skins/default/devices/default/`

If a localized file is not found during this process, the portal will perform a second search for the original non-localized file using the same search order. There should never be a case where no file is found. For a gadget to function, the non-localized file *must* be present.

Gadget Configuration Files

The portal requires gadget require to supply an XML file that describes their available settings. This information is used during the installation, configuration, and administration of the gadget. This file is named *AvailableSettings.xml* and will be stored under the portal's WEB-INF directory. This file uses DTDs to provide the strings used to describe the various settings.

The portal supports localized versions of these DTD files as well. DTD files should be localized with the same naming method used throughout this document. **Fig. 2** shows an example of a localized gadget.



The administration components of the portal that need access to these files will use the same routine provided by the portal to acquire localized versions of these files.